

Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the Application:

Listing of Claims:

1. (Previously Presented) A library system for creating programs executable on an industrial controller to control an industrial process, the library system comprising:

a library manager collecting in unique files, at least first and second program fragments having shared control variables determining physical inputs or outputs exchanged with the industrial process, the shared control variables having common tags;

a first program builder accepting user input to link in a first linking process instances of first program fragments from files in the library manager together to create a first portion of the control program; the first program builder renaming tags of control variables of duplicate instances of first program fragments to be unique; and

a second program builder accepting information about the first linking process, and user input, to create a second portion of the control program from second program fragments taken from the same files of the first program fragments used in the first portion of the control program, the second program builder renaming the tags of the control variables of the second program fragments to comport with the renaming of the tags of the control variables of the first portions by the first program builder;

whereby the second program fragments can communicate with the multiple instances of the first program fragments through common tags.

2. (Original) The library system of claim 1 wherein the first program fragments provide control logic for industrial control and the second program fragments provide visualization of industrial control.

3. (Original) The library system of claim 1 wherein the renaming of the first and second program fragments incorporate at least a portion of a name of their unique file of the library manager into the tags of the renamed first and second program fragments.

4. (Original) The library system of claim 1 wherein the library manager holds at least two first program fragments having shared control variables with a second program fragment.

5. (Original) The library system of claim 4 wherein the second program builder accepts user input to select from among the at least two first program fragments, a first program fragment with which the renaming of the tags of the control variable of the second program fragment will comport.

6. (Previously Presented) The library system of claim 5 wherein the second program builder provides at least one menu providing representations of first and second program fragments related to a common file of the library manager and wherein the user input for creating a second portion of the control program selects representations of the program fragments from the menu.

7. (Original) The development system of claim 6 wherein the menu depicts the first program fragments as dependent on particular items of physical equipment of the controlled process.

8. (Original) The library system of claim 1 wherein the library manager holds at least two second program fragments having shared control variables with a first program fragment.

9. (Original) The library system of claim 8 wherein the second program builder accepts user input to select from among the two second program fragments, a second program fragment with which the renaming of the tags of the control variables of the second program fragment to a first program fragment will comport.

10. (Previously Presented) The library system of claim 9 wherein the second program builder provides at least one menu providing representations of first and second program fragments related to common files of the library manager and

wherein the user input for creating a second portion of the control program selects representations of the program fragments from the menu.

11. (Original) The development system of claim 10 wherein the menu depicts the first program fragments as dependent on particular items of physical equipment of the controlled process.

12. (Original) The library system of claim 1 wherein the files of the library manager denote phases of operation of a machine of the controlled process and wherein the files also include information related to the phase of operation denoted by the file but not a program fragment.

13. (Previously Presented) The library system of claim 1 wherein the first program fragments are written in a language selected from the group consisting of: function block language, structured text language, ladder logic language and sequential function chart language.

14. (Original) The library system of claim 1 wherein the renaming is performed by concatenating a unique identifier onto the tag of the control variable.

15. (Original) The library system of claim 1 wherein the files of the library manager are identified to equipment of the controlled process.

16. (Previously Presented) A library system for creating programs executable on an industrial controller to control an industrial process, the library system comprising:

a library manager collecting in unique files, first and second program fragments having shared control variables representing physical inputs or outputs exchanged with the industrial process, the shared control variables having common tags;

a first program builder accepting user input to link in a first linking process instances of first program fragments from files in the library manager together to create a first portion of the control program; the first program builder renaming tags of control variables of duplicate instances of first program fragments to be unique; and

a second program builder accepting information identifying the files of the library manager from which the instances of the first program fragments originated to display to a user second program fragments related to each instance of the first program fragments according to common library files, and accepting user input to select among the displayed second program fragments to create a second portion of the control program from second program fragments, the second program builder renaming the tags of the control variables of the second program fragments to comport with the renaming of the tags of the control variables of the first portions by the first program builder;

whereby second program fragments that can communicate with the multiple instances of the first program fragments through common tags are identified and utilized.

17. (Original) The library system of claim 16 wherein the first program fragments provide control logic for industrial control and the second program fragments provide visualization of industrial control.

18. (Original) The library system of claim 16 wherein the renaming of the first and second program fragments incorporate a common name of their unique file of the library manager.

19. (Previously Presented) A system for creating a program, the system comprising:

a library having two pairs of program fragments stored in two library subportions, wherein one of the program fragments of each pair is of a first type and another of the program fragments of each pair is of a second type; and

a device capable of distinguishing both among the pairs of program fragments and among the program fragments of different types;

wherein the program is created in at least first and second stages during which, respectively, the program fragments of the first and second types are instantiated; and

wherein, during the second stage, the device identifies the program fragments of the second type that correspond to the program fragments of the first type that were instantiated during the first stage, based upon the library subportions in which the program fragments are stored.

20. (Previously Presented) The system of claim 19, wherein the instantiation of each of the program fragments includes modifying at least one variable associated with the respective program fragment so that the variable reflects an entity with which the program fragment has been associated in accordance with a user command.

21. (Previously Presented) The system of claim 20, wherein the instantiation of the program fragments of the second type includes modifying the variables associated with the respective program fragments of the second type so that the variables respectively conform to the respective variables of the respective program fragments of the first type that were already instantiated.

22. (Cancelled)

23. (Currently Amended) ~~The method of claim 22, further comprising:~~ A method of creating a program comprising:

providing a library storing a plurality of program fragments respectively within a plurality of library subportions, wherein at least some of the library subportions include a program fragment of a first type and a program fragment of a second type;

receiving commands to link instances of the program fragments of the first type with one another to form a control program;

instantiating the program fragments of the first type about which the commands were received, wherein the instantiating includes modifying variables of those program fragments;

identifying program fragments of the second type corresponding to the instantiated program fragments of the first type based upon the library subportions within which the program fragments are stored; and

instantiating the program fragments of the second type, wherein the instantiating includes modifying additional variables of those program fragments so that those variables conform with the variables of the corresponding instantiated program fragments of the first type.

24. (Previously Presented) The method of claim 23, wherein the program fragments of the first type provide control logic for industrial control and the program fragments of the second type provide visualization of industrial control.

25. (Previously Presented) A method of creating a program comprising:
providing a library storing a plurality of program fragments stored within a plurality of library subportions;
providing a model having hierarchically-ordered entities each of which respectively represents at least one of a component of a system and a process of the system;
receiving commands to associate at least some of the program fragments with the hierarchically-ordered entities; and
instantiating the program fragments about which the commands were received, wherein the instantiating of the program fragments automatically results in establishment of operational connections among the program fragments associated with the different entities.

26. (Previously Presented) The method of claim 25, wherein the instantiating includes modifying variables of the program fragments in a manner signifying the entities with which the program fragments have been associated.

27. (Previously Presented) The method of claim 26,
wherein at least some of the library subportions include program fragments of both first and second types, and
wherein when first and second program fragments of the first and second types, respectively, are instantiated in association with the same entity, the variables of the first and second program fragments are modified in a common manner that allows for operational connections to be achieved between those first and second program fragments.

28. (Previously Presented) A method of creating a program for execution on at least one industrial controller for controlling an industrial process, the method comprising:

providing a control program formed from a plurality of primary program fragments that have been instantiated, wherein the instantiated primary program fragments are respectively associated with respective process components that can be represented by elements in a model;

identifying a plurality of secondary program fragments stored within a plurality library subportions of a library that correspond to the instantiated primary program fragments within the control program, wherein correspondences between the program fragments are determined based upon whether the primary and secondary program fragments are stored within the same library subportions; and

instantiating at least some of the plurality of secondary program fragments so that the instantiated secondary program fragments are operationally connected to the instantiated primary program fragments.

29. (Previously Presented) The method of claim 28, wherein the instantiating includes modifying variables of the secondary program fragments in a same manner as corresponding variables of the corresponding primary program fragments were modified to arrive at the control program.

30. (Previously Presented) The method of claim 28, further comprising:
displaying the secondary program fragments stored within at least some of the library subportions that correspond to the instantiated primary program fragments; and
receiving user inputs selecting at least some of the secondary program fragments, which causes the instantiating of those secondary program fragments.

31. (Previously Presented) A library system for creating programs executable on an industrial controller to control an industrial process, the library system comprising:
a library manager collecting in unique files, at least first and second program fragments having shared control variables determining physical inputs or outputs exchanged with the industrial process, wherein the first program fragments are control logic program fragments that provide control logic for industrial control and the second program fragments are human machine interface (HMI) program fragments that enable visualization of industrial control, and wherein the shared control variables have common tags;

a first program builder accepting user input to link in a first linking process instances of first program fragments from files in the library manager together to create a first portion of a control program, wherein the first portion is a control logic portion of the control program, and wherein the first program builder renames tags of control variables of duplicate instances of the first program fragments to be unique; and

a second program builder accepting information about the first linking process, and user input, to create a second portion of the control program from second program fragments taken from the files from which the first program fragments used in the first portion of the control program were taken, wherein the second portion is a HMI program portion of the control program, and wherein the second program builder renames the tags of the control variables of the second program fragments to comport with the renaming of the tags of the control variables of the first portion by the first program builder;

whereby the second program fragments can communicate with the multiple instances of the first program fragments through common tags.